

## 9. Riley

**Program Name:** Riley.java

**Input File:** riley.dat

Riley's friend Priscella warned her about past programming challenges that required working with 2-D arrays in a non-standard pattern instead of straight rows and columns. Priscella suggested Riley practice by walking around the outer edge of the array and working inward.

The following example is NOT meant to show any pattern in the way that random data will actually exist in the table, the data in the table is specifically organized to only show the pattern of processing the cells in the table. Teams are NOT to infer any meaning; it is just data in a table that will be processed in a unique pattern.

Start at the top left corner and walk straight down the first column, visiting cells with 1 ... 7. Movement then changes direction and walks across the bottom row, visiting cells with 8 ... 13. Movement again changes direction and walks up the right column, visiting cells with 14 ... 19. Movement again changes direction and across the top row, visiting cells with 20 ... 24. The top cell of that column was previously visited and is not included. That same pattern is repeated with the remaining portion of the array, visiting cells with 25 ... 29, then 30 ... 33, then 34 ... 37, and 38 ... 40. The process continues, spiraling inward, until all cells are visited: cells with 41 ... 43, then 44 ... 45, then 46 ... 47, then 48, and finally 49.

Col → ↓ Row	0	1	2	3	4	5	6
0	1	24	23	22	21	20	19
1	2	25	40	39	38	37	18
2	3	26	41	48	47	36	17
3	4	27	42	49	46	35	16
4	5	28	43	44	45	34	15
5	6	29	30	31	32	33	14
6	7	8	9	10	11	12	13

For each path around the array as described above, calculate and output the average of the cell values visited during that trip around the array as shown below in the first line of sample output.

**Input:** First line contains a single integer **T** the number of test cases that follow with  $T \leq 10$ . Each test case starts with a line containing 2 integers separated by whitespace: **R**, the number of rows, and **C**, the number of columns, with both  $2 \leq R, C \leq 15$ . That line will be followed by **R** lines of data with each containing **C** integers separated by whitespace containing integers in  $[-100, 100]$ .

**Output:** For each test case, output 1 row of averages produced from the segments visited, separated by single space. A trailing space at end of each line is permitted. Display the averages with 2 digits after the decimal point.

**Sample input:**

```
3
7 7
1 24 23 22 21 20 19
2 25 40 39 38 37 18
3 26 41 48 47 36 17
4 27 42 49 46 35 16
5 28 43 44 45 34 15
6 29 30 31 32 33 14
7 8 9 10 11 12 13
```

~ Sample input continues on right ~

**Sample input continued**

```
3 6
40 90 -42 21 97 31
21 -28 -84 67 -85 -67
-30 -55 -36 -99 35 -22
5 2
11 -81
26 86
71 -23
-68 6
-62 48
```

**Sample output:**

```
12.50 32.50 44.50 49.00
-1.14 -32.50
1.40
```